Model-based and Data-driven Detectors for Time Synchronization Attacks against PMUs

Ezzeldin Shereen and György Dán
Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm, Sweden

***Abstract*—Precise time synchronization of Phasor Measurement Units (PMUs) is critical for monitoring and control of smart grids. Thus, time synchronization attacks (TSAs) against PMUs pose a severe threat to smart grid security. In this paper we present an approach for detecting TSAs based on the interaction between the time synchronization system and the power system. We develop a phasor measurement model and use it to derive an accurate closed form expression for the correlation between the frequency adjustments made by the PMU clock and the resulting change in the measured phase angle, without an attack. We then propose one model-based and three data-driven TSA detectors that exploit the change in correlation due to a TSA. Using extensive simulations, we evaluate the proposed detectors under different strategies for implementing TSAs, and show that the proposed detectors are superior to state-of-the-art clock frequency anomaly detection, especially for unstable clocks.**

***Index Terms*—PMU, time synchronization attacks, phasor measurements, security, cyber-physical systems, correlation, clock synchronization, data-driven detection, machine learning.**

I. Introduction

Measurements taken by Phasor Measurement Units (PMUs) have been employed in many smart grid applications in the past years (e.g., power oscillation damping, phase angle separation monitoring, and power system state estimation). For the reliable operation of these applications, the measurements taken by PMUs should be accurate, which requires precise time synchronization. Traditionally, time synchronization for PMUs is achieved through space based (e.g., GPS) or through network based time synchronization (e.g., PTPv2 [1]). However, both solutions have been shown to be vulnerable to time synchronization attacks (TSAs), both in theory and in practice. The signal from a GPS satellite or a PTP grandmaster can be spoofed [2], [3]. Moreover, PTPv2 is also vulnerable to the manipulations of the delays [4], which does not require breaking any cryptographic protection [5]. TSAs can have detrimental impact on the stability of the power grid [6], [7], thus it is of utmost importance to mitigate and detect them.

Existing methods that could be used to detect TSAs against GPS or PTPv2 depend on either information from the time synchronization system [8], [9] and the PMU clock [10], [11], or on information from the underlying power system (measured synchropasors) [12]–[14]. In this paper, we argue that combining information from both the PMU clock and the power system can significantly improve the detection accuracy of TSAs. We propose a model for the interaction between the PMU clock and the measured phase angle of the phasor. Based on the model, we derive a closed form expression for the correlation between changes in the PMU clock and resulting changes in the measured phase angle, and we show that the expression is accurate for a wide range of realistic clocks. We then propose four TSA detectors that analyze data from both the PMU clock and the power system; (1) a threshold-based detector based on the model, (2) a threshold-based data-driven detector, (3) a data-driven detector based on unsupervised machine learning, and (4) a data-driven detector based on supervised machine learning. The proposed detectors are shown to perform better than Cumulative Sum (CUSUM) [15], which is a state-of-the-art change detection method, for a wide range of possible variants of TSAs. Our results also show that the two low-complexity threshold-based detectors can achieve comparable performance to computationally extensive machine learning algorithms.

The rest of the paper is organized as follows. In Section II, we review the related work in TSA detection. We then introduce the phase angle model of the synchrophasor measurement in Section III. In Section IV, we analyse the model to estimate the correlation between the phasor measurement and the clock frequency adjustments, and validate this analysis. In Section V, we present the proposed TSA detectors. We then present the considered attack strategies, along with the performance of the proposed detectors in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

Previous work on detecting TSAs can be divided into three main categories: (1) mitigation and detection of GPS spoofing, (2) mitigation and detection of attacks on PTP, and (3) detection of clock anomalies. For GPS spoofing detection, authors in [8] exploited the doppler shift between a GPS transmitter and a receiver, while authors in [9] proposed a method based on the direction of arrival (DOA) of the signal compared to an expected value predicted by a Kalman filter. Most recently, authors in [16] proposed GPS time verification utilizing the known positions of the GPS receivers. For network time synchronization, [17] proposed a game theoretical framework for modeling the interaction between an attacker and a defender in a network using PTP. As mitigation, links suspected of malicious activity are put in "quarantine mode" in order to investigate the cause of the anomaly. Moreover, [3] proposed countermeasures to common security vulnerabilities

of PTP. The authors also proposed an extension of the PTP standard, including the use of efficient elliptic-curve public-key signatures for message authentication.

Several recent works considered the problem of clock frequency anomaly detection [10], [11]. Authors in [10] proposed a technique for anomaly detection in atomic clocks using the Dynamic Allan Variance (DAVAR), which is a measure of clock stability. Furthermore, a recursive frequency jump detector based on double exponential smoothing was presented in [11]. The detector takes into account the frequency drift, and detects changes in both drift and variance. Nevertheless, these techniques are based on variables that are not easily observable in PMU clocks (e.g., the clock frequency).

One common characteristic of the above approaches is that they all have considered the problem from the side of the time synchronization system and the clock. Another possible approach for the detection of TSAs could be utilizing information from the power system through PMU-based Linear State Estimation (LSE) [18]. If the deployed PMUs allow for observability of the power system, then the measurements taken by the PMUs could be used to estimate the state of the system, and to identify corrupted data using Bad Data Detection (BDD) techniques [18]. BDD techniques were originally developed for detecting faulty and noisy measurements. Since then, many research articles have proposed enhancing and tailoring BDD techniques to detect TSAs [19] and even to correct the malicious measurements [20]. Nonetheless, recent research has shown that TSAs could be designed to be undetectable by BDD [7], [21], [22]. Similarly, [12] has considered monitoring the correlations between measurements of electrically close PMUs to detect spoofing attacks (not necessarily TSAs). Other works have considered applying unsupervised [13] and supervised [14] machine learning algorithms on the power flow measurements in the power grid to detect spoofing attacks. Nonetheless, these works have not considered monitoring variables from the PMU clock system. One additional advantage of our approach is that it does not depend on information from multiple PMUs, and can therefore pinpoint exactly which PMU is attacked.

All existing approaches depend on either information from the time synchronization system and the clock, or on information from the power system (measured synchrophasors). To the best of our knowledge, no previous research has proposed leveraging information from both the clock and the power system to detect TSAs. In [23], we showed that combining information from the PMU clock and the power system can significantly improve the detection accuracy of TSAs. We introduced a model for the measured phase angle of synchrophasor measurements, and used it to derive an analytical result for the correlation between the measured phase angle and the periodical frequency adjustments made by the PMU clock. We then utilized this correlation to develop a threshold-based TSA detector. In this paper, we extend the work done in [23] by (1) extending the analytical correlation result to a variety of realistic clocks, (2) proposing and evaluating two additional detectors based on data analytics and machine learning, and (3) considering different strategies that an attacker can follow to implement its attack and evaluating their effect on the detection performance.

The efficient detection of TSAs is not only beneficial to the reliability of phasor measurements in smart grid applications, but would also be beneficial for other time synchronization-dependent applications. For example, in sensor networks deployed for object tracking, the sensors are required to be accurately time synchronized to estimate the object location [24]. A TSA would change the time perceived by the sensor, and thus the estimation of the location would be inaccurate. Another example of time-sensitive applications is in the growing field of collaborative robots, which also rely on precise time synchronization [25]. We believe that the ideas presented in this paper could be inspiring for developing TSA detectors for those time-sensitive applications, by monitoring data from both the time synchronization system and the collected physical quantities.

III. System Model

We consider a PMU that periodically measures a voltage or current phasor. The true phase angle measured in radians at time instant t_i is denoted by $\alpha_p(t_i)$, and the corresponding zero crossing time lag w.r.t. a reference signal is denoted by $O_p(t_i)$, where $\alpha_p(t_i) = 2\pi f O_p(t_i)$, and f is the nominal oscillation frequency of current and voltage signals in the system. In the following we express all quantities in terms of time lag. The measured phase angle is dependent not only on the true phase $O_p(t_i)$ but also on the accuracy of the metering device and the accuracy of the PMU clock. In what follows we present models for these factors.

A. Process and Measurement Model

We adopt a widely used measurement model in power systems that assumes that the phase angle, and hence $O_p(t_i)$ follows a random walk [26]. Thus, $O_p(t_i) = O_p(t_{i-1}) + \omega_p(t_i)$, where $\omega_p(t_i) \sim \mathcal{N}(0, \sigma_p)$ is zero mean normally distributed process noise. Furthermore, the measurement error can be modelled by additive white gaussian noise $\omega_n(t_i) \sim \mathcal{N}(0, \sigma_n)$ [26]. Therefore, the measured phase angle for a perfectly synchronized PMU clock can be expressed as

$$\begin{aligned} O_z(t_i) &= O_p(t_i) + \omega_n(t_i) \\ &= O_z(t_{i-1}) + \omega_z(t_i), \end{aligned} \tag{1}$$

where $\omega_z(t_i) = \omega_p(t_i) + \omega_n(t_i) - \omega_n(t_{i-1}) \sim \mathcal{N}(0, \sigma_z)$, and $\sigma_z = \sqrt{\sigma_p^2 + 2\sigma_n^2}$. Note that $\omega_z(t_i)$ is a sequence of identically distributed, but not independent Gaussian random variables.

B. Clock Offset Model

When the PMU clock is not perfectly synchronized, the measured phase angle $O_m(t_i)$ can be given by

$$O_m(t_i) = O_z(t_i) + O_c(t_i), \tag{2}$$

where $O_c(t_i)$ denotes the time offset between the PMU clock and the correct time. $O_c(t_i)$ depends on two factors: (1) the

accuracy of the PMU clock determined by the clock frequency drift, and (2) the synchronization mechanism that adjusts the PMU clock frequency based on a time reference.

1) Clock Drift Model: We model the clock frequency deviation, denoted by $\gamma(t)$, by an Ornstein-Uhlenbeck (OU) process, which was shown to be a suitable model for this purpose [27]. The OU process is a stationary Gauss-Markov process, and is given in the discrete and approximated form by the Euler-Marugama method [28] as

$$\gamma(t_i) = \gamma(t_{i-1}) + \theta(\mu - \gamma(t_{i-1}))(t_i - t_{i-1}) + \omega_\gamma(t_i)\sqrt{t_i - t_{i-1}}, \quad (3)$$

where μ is the long term mean of the process, $\theta > 0$ is the speed of reversion, which determines how fast $\gamma(t)$ drifts to μ, $\omega_\gamma(t_i) \sim \mathcal{N}(0, \sigma_\gamma)$, and $t_i - t_{i-1}$ is the time duration between samples. For example, in PTP $t_i - t_{i-1} = 1$ seconds, which is the time duration between synchronization messages. Therefore, without loss of generality, in what follows we replace t_i and t_{i-1} by t and $t-1$, respectively. Moreover, for simplicity we assume that $\gamma(0) = \mu$. Therefore, (3) becomes

$$\begin{aligned} \gamma(t) &= \gamma(t-1) + \theta(\gamma(0) - \gamma(t-1)) + \omega_\gamma(t) \\ &= \theta\gamma(0) + (1-\theta)\gamma(t-1) + \omega_\gamma(t), \end{aligned} \quad (4)$$

i.e., a weighted average of the long term mean $\gamma(0)$ and the previous value $\gamma(t-1)$ in addition to random noise $\omega_\gamma(t)$.

2) Clock Servo and Adjustments: Typically, PMU clocks are synchronized to a time reference by a component called the clock servo. The function of the clock servo is to adjust the clock frequency to minimize the raw offset $\hat{O}_c(t) = \tau_{pmu}(t) - \tau_r(t)$ between the time perceived by the PMU $\tau_{pmu}(t)$ and the reference time $\tau_r(t)$, in a smooth and gradual manner. The clock servo is usually implemented as a P-controller or a PI-controller, such as the open source PTP implementation PTPd [29]. We consider the general case of a PI-controller, defined by two parameters; a proportional gain K_p and an integrator gain K_i. Typical values are $K_p = 0.1$ and $K_i = 0.001$ [29]. The frequency adjustment is then computed based on the raw offset $\hat{O}_c(t)$ as

$$A(t) = D(t) + K_p\hat{O}_c(t), \quad (5)$$

where $D(t)$ is the accumulated integrator error of the PI-controller given by $D(t) = D(t-1) + K_i\hat{O}_c(t)$.

The true offset $O_c(t)$ between the PMU clock and the correct time is dependent on both the frequency deviation $\gamma(t)$ and the introduced adjustments $A(t)$ according to

$$O_c(t) = O_c(t-1) + \int_{t-1}^{t} \gamma(t)\,dt - A(t-1). \quad (6)$$

Note that the computed raw offset $\hat{O}_c(t)$ is an estimation of the true offset $O_c(t)$, but the two are not necessarily equal (e.g. in case of a TSA). Figure 1 summarizes the factors affecting the measured phase angle O_m by a PMU.

C. Attack Model

We consider an attacker that is able to manipulate a time synchronization source (e.g., spoof a GPS signal or manipulate a PTP synchronization message), and hence change the reference time sent by the source (c.f. Figure 1). Such manipulations are realistic as shown in many works, such as [2] and [4]. We denote by t_s^a and t_e^a the start and end time of the attack, respectively. The attacker can generate fake time references $\tau_r^a(t)$ for $t \in [t_s^a, t_e^a]$, which will affect the measured raw offsets $\hat{O}_c^a(t) = t_{pmu}(t) - \tau_r^a(t)$.

Our model does not assume that the attacker knows the clock servo parameters $(K_p, K_i, D(t))$. In principle the attacker could know these parameters, as most clock servo implementations are open source or vulnerable to reverse engineering, and this knowledge could allow the attacker to compute the effect of $\hat{O}_c^a(t)$ on the adjustments $A^a(t)$ through (5). Nevertheless, knowledge of the clock servo parameters is not necessary to launch the attacks described in this paper.

Our attack model considers a powerful attacker, but is in accordance with Kerckhoff's priniple, and allows to identify information that need to be protected cryptographically in order to make the system secure.

IV. Model-assisted Estimation of the Correlation

Our proposed TSA detectors are based on analyzing data from the PMU clock and the power system. This is achieved through monitoring the correlation between the PMU clock and the phase angle of the measured phasor, leveraging that this correlation would be affected by an attack. Therefore, in the following we set out to obtain a closed-form expression for the expected value of the correlation between the clock frequency adjustments $A(t-1)$ made by the PMU clock servo at time step $t-1$, and the resulting change in the phase angle of the measured phasor at the following time step, denoted by $\Delta O_m(t) = O_m(t) - O_m(t-1) = \Delta O_z(t) + \Delta O_c(t)$. We then evaluate the accuracy of this closed-form expression in estimating the measured correlation under different model parameters.

A. Correlation Analysis

In what follows we present an approximate analysis of the correlation between $A(t-1)$ and $\Delta O_m(t)$ without an attack. To make the analysis tractable, we consider that the clock servo uses a P-controller instead of a PI-Controller, thus $K_i = 0$ and $D(t) = 0, \forall t$. Unlike the analysis in [23], which assumes $\theta = 1$, our analysis is valid for any $\theta > 0$ and it thus applies to practical clocks, which can typically be modelled by very small values of θ. In Section IV-B and IV-C we will quantify the effect of our assumption on the accuracy of the analysis.

Proposition 1. *Consider that $K_i = 0$. Then the correlation $\rho_{(\Delta O_m(t), A(t-1))}$ between $\Delta O_m(t)$ and $A(t-1)$ can be approximated as*

$$\begin{aligned} \tilde{\rho}_{(\Delta O_m(t), A(t-1))} &= \frac{cov((\Delta O_m(t), A(t-1)))}{\sigma_{\Delta O_m}\sigma_A} \\ &= \frac{E[(\Delta O_m(t) - \mu_{\Delta O_m})(A(t-1) - \mu_A)]}{\sigma_{\Delta O_m}\sigma_A}, \end{aligned} \quad (7)$$

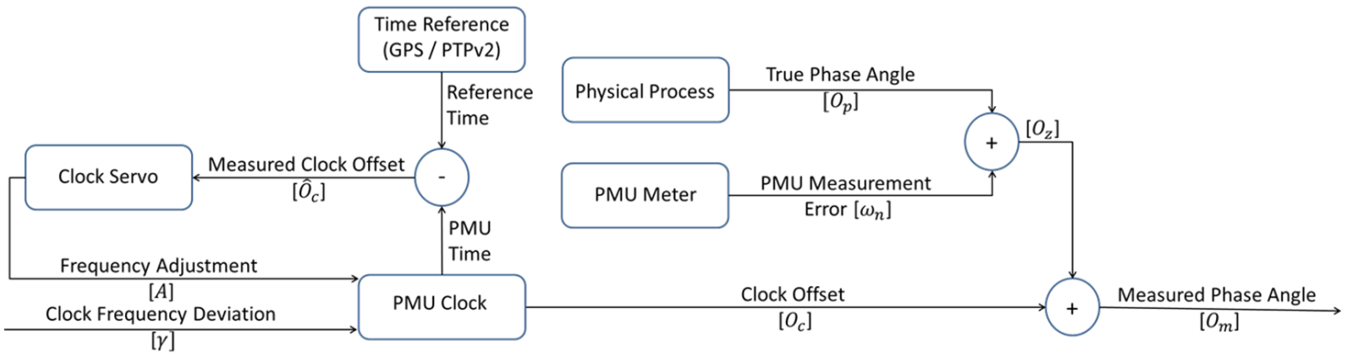


Fig. 1: Block Diagram of the variables affecting the phase angle measured by a PMU.

where $cov((\Delta O_m(t), A(t-1))), \sigma_{\Delta O_m}$, *and* σ_A *are given in* (8)*, with* $\delta = 1 - K_p$*,* $\beta = 1 - \theta$*,* $\varphi = \frac{\beta+1}{\beta-\delta}$*, and* $f_g(x,t) = \frac{1-x^{t-2}}{1-x}$.

Proof. Observe that for $K_i = 0$ the frequency adjustments can be computed as $A(t) = K_p \hat{O}_c(t)$. In the absence of TSAs, one can assume that $O_c(t) = \hat{O}_c(t)$, and hence $A(t) = K_p O_c(t)$. Furthermore, we can approximate (6) using linear interpolation between $t-1$ and t to obtain

$$O_c(t) = (1 - K_p) O_c(t-1) + \frac{\gamma(t) + \gamma(t-1)}{2}. \quad (9)$$

This approximation is reasonable since γ is not expected to change rapidly during one second. Now, assuming $A(0) = O_c(0) = 0$, we can rewrite (9) to get

$$\begin{aligned} O_c(t) &= \left(\sum_{k=1}^{t-1} \left[\delta^{t-k} + \varphi(\beta^{t-k} - \delta^{t-k}) \right] * \frac{\omega_\gamma(k)}{2} \right) \\ &+ \frac{\omega_\gamma(t)}{2} + \sum_{k=0}^{t-1} \delta^k \gamma(0). \end{aligned}$$

For very large t, the last term converges to $\frac{\gamma(0)}{K_p}$. Additionally, we can express

$$\begin{aligned} \Delta O_c(t) &= \left(\sum_{k=1}^{t-2} \left[\varphi \left([\beta^{t-k} - \beta^{t-k-1}] - [\delta^{t-k} - \delta^{t-k-1}] \right) \right. \right. \\ &+ \left. \left. \delta^{t-k} - \delta^{t-k-1} \right] * \frac{\omega_\gamma(k)}{2} \right) \\ &+ (\delta + 1 + \varphi[\beta - \delta]) \frac{\omega_\gamma(t-1)}{2} + \frac{\omega_\gamma(t)}{2}. \end{aligned} \quad (10)$$

Using $\Delta O_m(t) = \Delta O_c(t) + \omega_z(t)$ and $A(t-1) = K_p O_c(t-1)$ in (7), and using the fact that $E[\omega_\gamma(i)\omega_\gamma(j)] = 0$ for $i \neq j$, and that $\mu_{\Delta O_m} = 0, \mu_A = \gamma(0)$, after some algebraic manipulation we obtain (8), which proves the result. □

Observe that the expression for $\tilde{\rho}(\Delta O_m(t), A(t-1))$ based on (8) does not depend on the individual values of σ_z and σ_γ. It depends however on their ratio $\sigma^* = \frac{\sigma_z}{\sigma_\gamma}$. Therefore, we can write $\tilde{\rho}_{(\Delta O_m(t), A(t-1))} = f_\rho(t, \sigma^*, K_p, \theta)$. Observe also that if the PMU is attacked, then $O_m(t) \neq \hat{O}_m(t)$, and the value of the correlation is expected to significantly change. In what follows we evaluate the accuracy of the proposed correlation analysis and the obtained closed form expression $f_\rho(t, \sigma^*, K_p, \theta)$.

B. Validation of the Correlation Analysis for $K_i = 0$

To assess the accuracy of $f_\rho(t, \sigma^*, K_p, \theta)$ for $K_i = 0$, we considered a PMU clock with a P-controller clock servo ($K_i = 0$). The results reported are the averages of 5000 simulations of the measured phase angle of a PMU according to the previously mentioned measurement and clock models. Figure 2 shows the result of the closed form expression $f_\rho(t, \sigma^*, K_p, \theta)$ and the computed empirical correlation $\rho_N(t)$ from the simulations, as a function of σ^* for various values of the correlation window size N and the speed of reversion θ.

The figure shows that the correlation changes very little when σ^* is either very high or very low. Therefore, $f_\rho(t, \sigma^*, K_p, \theta)$ would be insensitive to errors in estimation of σ^* in these regions. In fact, when θ is small ($\theta = 0.01$), the value of the correlation hardly changes for any σ^*. The figure also shows that the accuracy of $f_\rho(t, \sigma^*, K_p, \theta)$ increases when N increases. The reason for the discrepancy between $f_\rho(t, \sigma^*, K_p, \theta)$ and the empirical correlation $\rho_N(t)$ is due to the difference between time averages and ensemble averages. $\rho_N(t)$ measures the correlation over time between a sequence of N pairs of random variables. On the contrary $f_\rho(t, \sigma^*, K_p, \theta)$ is the correlation between two variables; $\Delta O_m(t)$ and $A(t-1)$ at one time instant. To obtain the equivalent of $f_\rho(t, \sigma^*, K_p, \theta)$ numerically, one would have to compute the correlation between $\Delta O_m(t)$ and $A(t-1)$ across multiple simulations. To show that this is indeed the case, Figure 2 also shows $\rho_{runs}(t)$, which is the correlation computed across the 5000 simulations. The figure shows that this ensemble average correlation is an excellent match for $f_\rho(t, \sigma^*, K_p, \theta)$ as expected. Observe that $\rho_{runs}(t)$ can not be computed in practice, as it is impossible to run multiple copies of the same time synchronization system, but $\rho_N(t)$ can be computed efficiently. Overall, the results show that $f_\rho(t, \sigma^*, K_p, \theta)$ is a good approximation of $\rho_N(t)$ for large enough window size N.

$$
\begin{aligned}
cov((\Delta O_m(t), A(t-1))) &= K_p\Big(\frac{\sigma_\gamma}{2}\Big)^2 \big[\, f_g(\delta^2,t)\left((1-2\varphi+\varphi^2)(\delta^3-\delta^2)\right) + f_g(\beta^2,t)\left(\varphi^2(\beta^3-\beta^2)\right) \\
&+ f_g(\delta\beta,t)\left((\varphi-\varphi^2)(\delta\beta^2-2\delta\beta+\delta^2\beta)\right) + \delta - \varphi\delta + \varphi\beta - 1\,\big] \\
\sigma^2_{\Delta O_m} &= \Big(\frac{\sigma_\gamma}{2}\Big)^2 \big[f_g(\delta^2,t)\left((\delta^4+\delta^2-2\delta^3)(1+\varphi^2-2\varphi)\right) + f_g(\beta^2,t)\left(\varphi^2(\beta^4+\beta^2-2\beta^3)\right) \\
&+ f_g(\delta\beta,t)\left(2(\varphi-\varphi^2)(\delta^2\beta^2-\delta^2\beta-\delta\beta^2+\delta\beta)\right) + (\delta+\varphi\beta-\varphi\delta-1)^2+1\big] \\
\sigma^2_A &= \Big(\frac{K_p\sigma_\gamma}{2}\Big)^2 \big[f_g(\delta^2,t)\left(\delta^2(1+\varphi^2-2\varphi)\right) + f_g(\beta^2,t)\left(\varphi^2\beta^2\right) + f_g(\delta\beta,t)\left(2\delta\beta(\varphi-\varphi^2)\right)+1\big]
\end{aligned}
\tag{8}
$$

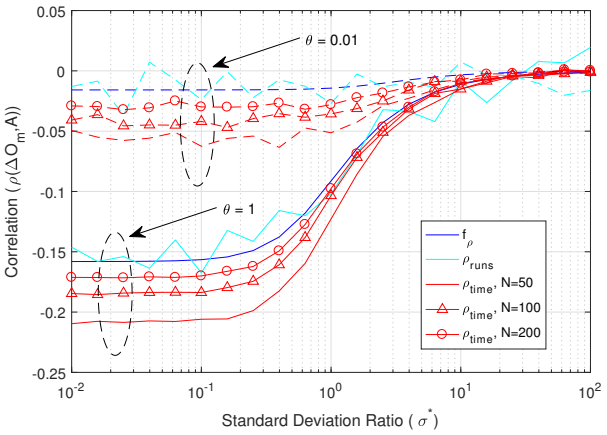


Fig. 2: Comparison of $f_\rho(t,\sigma^*,K_p,\theta)$, ρ_{runs}, and ρ_N vs. σ^*. $\theta=\{0.01,1\}$, $K_p=0.1$, and $t=2000$.

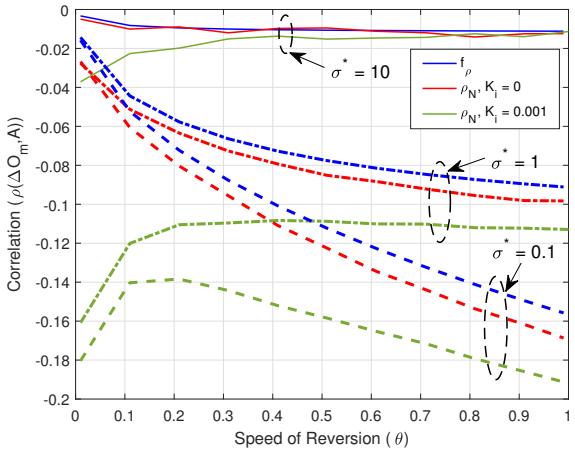


Fig. 3: Comparison of $f_\rho(t,\sigma^*,K_p,\theta)$, and ρ_N vs. θ. $\sigma^*=\{0.1,1,10\}$, $K_p=0.1$, $K_i=0.001$, $N=200$, and $t=2000$.

C. Validation of the Correlation Analysis for $K_i \geq 0$

To evaluate the sensitivity of $f_\rho(t,\sigma^*,K_p,\theta)$ to the controller integrator gain K_i, Figure 3 shows f_ρ and ρ_N as a function of θ, for various values of σ^*, both for $K_i=0$ (P-controller) and $K_i=0.001$ (PI-controller, as used in PTPd). For $K_i=0$ (P-controller), the figure shows that f_ρ is a very accurate approximation of ρ_N as previously concluded from Figure 2. For $K_i=0.001$ the results show that f_ρ is a reasonably good approximation, especially for high values of σ^*. Nevertheless, the approximation is not as accurate as in the case $K_i=0$ since the approximation error has two sources; (1) the approximation of the ensemble average with the time average (c.f., Fig 2), and (2) the error due to $K_i\neq 0$. We observe also that f_ρ is a better estimate of ρ_N for higher values of θ. Overall, we can conclude that f_ρ is close to the empirical correlation ρ_N despite the approximation.

V. Correlation-based TSA Detectors

In what follows we propose four different detectors for the detection of TSAs based on the analysis of the sequences of the clock frequency adjustments A and the measured phase angles O_m. The first detector, a model-based detector, follows and utilizes the correlation analysis presented in section IV-A, while the second detector is model-free. These two threshold-based detectors are based on computing the difference between the predicted value of the correlation $\hat{\rho}$ and the measured value. The third and fourth detectors are based on unsupervised and supervised machine learning classification algorithms, respectively.

A. Model-based Detector

In the model-based detector, we compute the predicted correlation $\hat{\rho}$ using the correlation analysis presented in Section IV-A. To do that, we estimate σ_γ, σ_z (hence σ^*), and θ based on previous knowledge regarding the PMU clock accuracy, and the power system stability. The predicted correlation can then be computed as $\hat{\rho}=f_\rho(t,\sigma^*,K_p,\theta)$ for a sufficiently high value of t, where K_p is a known clock servo parameter. We emphasize that the accuracy of the estimated σ^* and θ plays a crucial role in the accuracy of the predicted correlation $\hat{\rho}$. The empirical correlation $\rho_N(t)$, on the other hand, is computed for every time step t based on $A(\tau-1)$ and $\Delta O_m(\tau)$, $\tau\in\{t-N+1,\ldots,t\}$, where N is the correlation window length. An alarm is then raised at time t if $|\rho_N(t)-\hat{\rho}|>\eta$, where η is the detection threshold.

B. Model-free Detector

In the model-free detector the predicted correlation $\hat{\rho}$ is computed from the historical values of $A(t-1)$ and $\Delta O_m(t)$ from the actual system, when the PMU is known to be in a normal state (non-attacked). The computation of the empirical

correlation $\rho_N(t)$ and the generation of detection alarms follows the same procedure as the model-based detector.

C. Auto-Encoder Neural Network Detector

The Auto-Encoder Neural Network (AENN) detector [30] is an example of an unsupervised machine learning anomaly detector. For this detector, historical values of $A(t)$ and $\Delta O_m(t)$ are recorded, and are used to learn a neural network model for non-attacked data. An alarm is raised in real-time if the data obtained from the PMU does not fit the learned model well. In an AENN, the input and the output layers of the neural network have the same dimension, while the hidden layers have lower dimensions. The objective is then to learn a network that can encode (compress) the data, and then decode (decompress) it again with the least root-mean-squared error $(RMSE)$ between the input and the output layers. An alarm is raised if $RMSE > \eta$, where η is the detection threshold. The idea of AENNs is similar to Principal Component Analysis (PCA), but its performance is usually superior to PCA as it is capable of learning non-linear relations between the input and the output layers, while PCA is limited to linear relations.

D. Random Forest Detector

The Random Forest (RF) detector [31] is an example of a supervised machine learning detector, where historical values of $A(t)$ and $\Delta O_m(t)$ are recorded, including sequences of attacked and non-attacked data. The labelled data is then used to learn a model for classification between attacked and non-attacked sequences. An RF detector consists of a collection of decision trees; each tree learns a set of rules to differentiate between attacked and non-attacked sequences. Given an observed sequence of $A(t)$ and $\Delta O_m(t)$, the RF detector outputs a value $P_{attack} \in [0, 1]$, which is the probability that the sequence is attacked. P_{attack} is computed based on averaging the output of the individual trees. An alarm is then raised if $P_{attack} > \eta$, where η again is the detection threshold. We chose to use the RF detector to classify attacked and non-attacked data due to its simplicity, and due to its resistance to the over-fitting problem [32].

Unlike the model based detector (detector A), detectors (B-D) do not depend on the accuracy of the system model. However, they require access to historical data. For detectors B and C, the historical data have to be collected when the system is in a non-attacked state, while detector D requires historical data for both the attacked and the non-attacked state.

VI. Numerical Results

In this section we evaluate our proposed detectors based on synthetic data.

A. Attack Strategies

For the evaluation we consider an attacker that aims to achieve an offset $\Delta\tau^a = \tau_{pmu} - \tau_r$ between the PMU time and the reference time. We refer to this offset as the attack goal. Intuition says that an attack should become easier to detect as $\Delta\tau^a$ increases, but at the same time, the rate at which the attack is performed would likely affect the detection performance. To cover a wide variety of attack strategies, we consider a general attacker model, in which the attacker applies the attack for a duration $\Delta t^a = t_e^a - t_s^a$ until the target offset $\Delta\tau^a$ is reached. For every $t \in [t_s^a, t_e^a]$ we model the computed raw offset after an attack as $\hat{O}_c^a(t) \sim \mathcal{N}(\hat{O}_c(t) + \mu_a(t), \sigma_a(t))$, where $\mu_a \neq 0$, and $\hat{O}_c(t)$ represents the corresponding computed raw offset if no attack was present. This allows us to model an attacker that intends to accelerate $(\mu_a > 0)$ or decelerate $(\mu_a < 0)$ the clock. Note that the attack rate $\mu_a(t)$ needed to implement the attack will increase as the attack duration Δt^a decrease. Therefore, for a given attack goal $\Delta\tau^a$, an attack should be easier to detect as Δt^a decreases (hence μ_a increase).

In order to achieve her goal, we consider that the attacker increases $\mu_a(t)$ from 0 to the maximum attack rate μ_a^{max} by time t_1^a, and then it decreases $\mu_a(t)$ starting at time t_2^a. In the following we consider three different strategies for increasing and for decreasing $\mu_a(t)$. In section VI-C, we will investigate the effect of the chosen strategy on the attack detectability.

1) Rectangular Attack: In the rectangular attack strategy, $t_1^a = t_s^a$ and $t_2^a = t_e^a$, which means that $\mu_a(t)$ is set to a constant value $\mu_a(t) = \mu_a^{max} = \frac{\Delta\tau^a}{\Delta t^a}, \forall t \in [t_s^a, t_e^a]$.

2) Triangular Attack: In the triangular attack strategy, $t_1^a = t_2^a = \frac{t_s^a + t_e^a}{2}$, and $\mu_a(t)$ is changed in a linear fashion given by

$$\mu_a(t) = \mu_a^{max}\left(1 - \frac{|t - t_1^a|}{\frac{\Delta t^a}{2}}\right), \forall t \in [t_s^a, t_e^a],$$

where $\mu_a^{max} = \frac{2\Delta\tau^a}{\Delta t^a}$.

3) Logistic Attack: In the logistic attack strategy, $t_1^a = t_s^a + \Gamma\Delta t^a$ and $t_2^a = t_e^a - \Gamma\Delta t^a$, where $\Gamma \in [0, 1]$ is a constant value that indicates the midpoint of the logistic curve w.r.t. Δt^a. Furthermore, $\mu_a(t)$ changes according to a logistic curve (which is smoother than a linear curve). Therefore, using $\mu_a^{max} = \frac{\Delta\tau^a}{\Delta t^a(1-\Gamma)}$, we can write $\mu_a(t)$ as

$$\begin{cases} \mu_a^{max}\left(\frac{1}{1+\exp(-\frac{50}{\Delta t^a}(t - t_s^a - \frac{\Gamma}{2}\Delta t^a))}\right) & t \in [t_s^a, t_1^a[\\ \mu_a^{max} & t \in [t_1^a, t_2^a[\\ 1 - \mu_a^{max}\left(\frac{1}{1+\exp(-\frac{50}{\Delta t^a}(t - t_s^a - \frac{1-\Gamma}{2}\Delta t^a))}\right) & t \in [t_2^a, t_e^a] \end{cases}.$$

Figure 4 shows the frequency adjustments $A(t)$ as computed by the clock servo when implementing the considered attack strategies. In all strategies the attack goal was set to $\Delta\tau^a = 1$ms and the attack duration was set to $\Delta t^a = 1000$ seconds (from $t_s^a = 600$s to $t_e^a = 1600$s). Observe that the adjustments do not follow exactly the shape of the attack strategy (rectangular, triangular, etc.) due to the smoothing effect of the PI-controller.

B. Evaluation Methodology

We consider two clocks: clock A is a very accurate clock characterized by $(\gamma(0) = 100ns, \sigma_\gamma = 10ns)$, while clock B is a less accurate clock characterized by $(\gamma(0) = 1\mu s, \sigma_\gamma = 100ns)$. The oscillator frequency of both clocks has a speed of reversion $\theta = 10^{-6}$, which is a reasonable value, as the

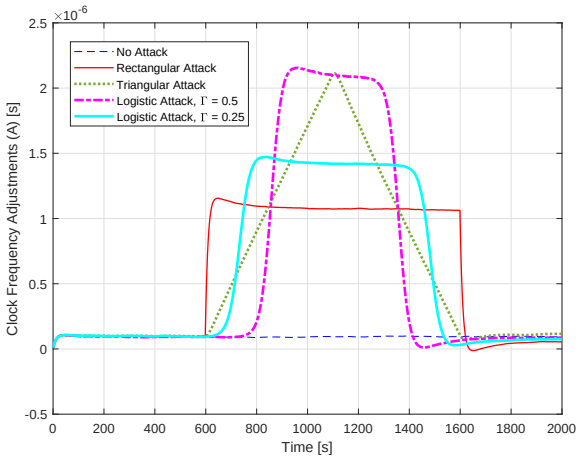


Fig. 4: Frequency Adjustments made to the PMU clock , when implementing the different attack strategies. $\Delta\tau^a = 1$ms, $\Delta t^a = 1000$s, $\gamma(0) = 100$ns, $\sigma_\gamma = 10$ns, $\sigma_z = 2.2\mu$s, $K_p = 0.1, K_i = 0.001$, and $\theta = 10^{-6}$.

frequencies of real clocks usually have weak, but non-zero tendencies to revert to the long term mean $\gamma(0)$. Furthermore, we consider that the clock servo for both clocks uses a PI-controller with $K_p = 0.1$ and $K_i = 0.001$. Regarding the PMU metering accuracy, we consider that $\sigma_z = 2.2\mu s$, which is a realistic value if the PMU uses a class 0.1 sensor. Therefore, we get $\sigma^* = 220$ and $\sigma^* = 22$ for clock A and B, respectively.

For both clocks we used the phase angle model in section III to generate both attacked and non-attacked sequences of $A(t)$ and $\Delta O_m(t)$. Each sequence is 2000 seconds long, and for the attacked sequences we used $t_s^a = 600$s and $\sigma_a(t) = \mu_a(t)/10$. We divided the generated sequences into different scenarios; each scenario is defined by (1) the attack goal $\Delta\tau^a$, (2) the attack duration Δt^a, and (3) the attack strategy. For each scenario, we recorded 1000 non-attacked sequences and 1000 attacked sequences. We refer to the collection of 2000 sequences (1000 non-attacked + 1000 attacked) of one scenario as a dataset.

In order to use the proposed detectors on the datasets, several detector parameters had to be set. For the model-based and the model-free detectors, a sliding window of length $N = 200$ was used to compute the correlations. For AENN, the input to the neural network was non-overlapping windows of length 50 seconds of the recorded non-attacked sequences of ΔO_m and A (in total 100 variables), while the dimension of the single hidden layer of the AENN was 50 (thus, a compression ratio of 0.5). Furthermore, we used the logistic sigmoid function as an activation function for the neurons of the AENN. For the RF detector, we also used non-overlapping windows of length 50 of the recorded sequences. Each trained forest consisted of 20 trees, with a maximum allowed tree depth of 40. Furthermore, we train one RF detector on the data from all attack scenarios (datasets). Training a separate detector for each dataset could possibly improve the detection performance, but is impractical. First, there is a vast number of potential attack scenarios. Second, it is unclear which detector should be used as the attacker's strategy is unknown a priori.

As a baseline we used CUSUM [15], which is a state-of-the-art clock frequency anomaly detector. To implement CUSUM over the adjustment sequences $A(t)$, we computed the mean adjustment $\bar{A}$ and the standard deviation s_A over the first 200 seconds of each sequence. Next, we compute the cumulative sum of the difference between the adjustments and $\bar{A}$ for each time instant t as $\sum_{k=1}^{t} A(k) - \bar{A}$. An alarm is then generated if at any time instant t, the cumulative sum exceeds ηs_A, where η is the detection threshold for CUSUM. We have also experimented with applying CUSUM on the correlation time series $\rho_N(t)$. Nevertheless, we observed that this performed slightly worse than the proposed model-free detector, and thus we do no not show corresponding results.

To compare the tested detectors, we utilize the area under the Receiver Operating Characteristic (ROC) curve [33]. The ROC curve shows the trade-off between the detection power of a detector and its resistance to false alarms. The x-axis of a ROC curve shows the false positive rate (FPR) of a detector, while the y-axis shows the true positive rate (TPR). FPR$\in [0, 1]$ is defined as the proportion of non-attacked sequences that gets classified by the detector as attacks. TPR$\in [0, 1]$ (also called the recall) is the proportion of attacked sequences that gets classified by the detector as attacks. Each point on the ROC curve is obtained by using a different value of the detection threshold η. Setting very low values of η makes the detector generate an alarm for many sample data, resulting in TPR and FPR very close to 1, which corresponds to the upper-right corner of the ROC curve. On contrary, very high values of η would not let the detector generate any alarms, thus resulting in TPR and FPR very close to 0, which corresponds to the lower-left corner of the ROC curve. The range of thresholds that should be used depends on the detector. For example, for the correlation-based detectors, the threshold corresponds to the difference between the expected correlation and the measured one ($|\rho_N(t) - \hat{\rho}|$). Since the correlation values lie in the range $[-1, 1]$, the difference between them cannot exceed 2. In practice, η is usually set to achieve a certain desired FPR.

The area under the ROC curve (AUC) is a widely accepted performance measure of detectors. If a threshold value η^* that classifies all attacked sequences as such (TPR=1), without generating any false alarms (FPR=0) exists, then AUC=1 and the detector is called a perfect detector. Another interesting case arises when AUC=0.5, which means that the evaluated detector does not perform better than random guessing. For more information regarding ROC curves and their interpretations, we refer to [33].

C. Detector Performance

Figure 5 shows the ROC curves obtained by CUSUM and the 4 proposed detectors for attack target $\Delta\tau^a = 100\mu s$, and attack duration $\Delta t^a = 100$ seconds, using the rectangular attack strategy against clock A. The values of the AUC are

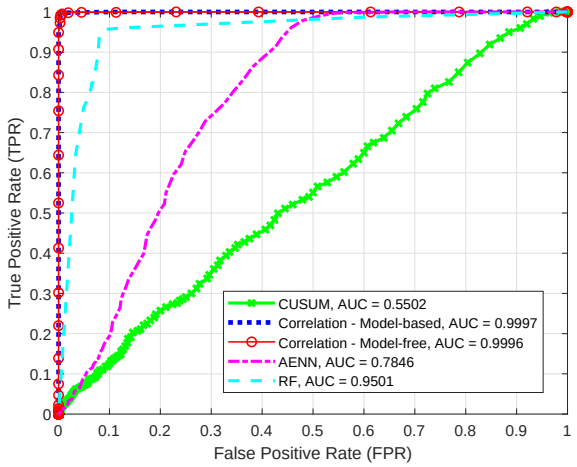


Fig. 5: ROC curves obtained by the five detectors, for clock A on the dataset defined by ($\Delta\tau^a = 100\mu$s, $\Delta t^a = 100$s, rectangular attack).

shown in the legends. The figure shows that for this scenario the model-based and the model-free detectors outperform all other detectors, even the RF classifier (which is trained on the attacked data). In particular, we see that the ROC curves for the model-based and the model-free detectors include points that are very close to the point (TPR=1, FPR=0). Furthermore, the CUSUM detector is far inferior to all other detectors. In fact, the performance of CUSUM is very close to a random detector. In order to gain a more comprehensive view on the performance of the detectors, in the following we show performance results on a variety of datasets, corresponding to different attack scenarios and different clocks.

Figure 6 shows the performance of CUSUM and the 4 proposed detectors for clock A (Figure 6a) and clock B (Figure 6b). Each plot shows an isoquant (contour curve) of the AUC obtained by each of the tested detectors, as a function of the attack duration Δt^a on the x-axis and the attack goal $\Delta\tau^a$ on the y-axis, using the rectangular attack strategy. To easily understand and compare the curves, recall that a detector performs better if the area covered by high AUC values is larger. The annotations show the AUC value for the dataset with $\Delta t^a = 200$s and $\Delta\tau^a = 100\mu$s, so as to facilitate the comparison of the detectors for this dataset. We chose this dataset as it clearly highlights the difference between the different detectors for the two clocks. From the figure, we can see several interesting observations. First, for most of the plots we see an expected pattern that the detectors perform better when Δt^a is small and when $\Delta\tau^a$ is high, as the attacker will need to increase the rate $\mu_a(t)$ in both cases. This pattern is not obvious for CUSUM, as the performance only depends on the attack goal $\Delta\tau^a$. Second, for all detectors, the performance is better for clock A than for clock B. This is also an expected result because clock A is a more accurate (and stable) clock than clock B, and the change arising due to the attack should be easier to detect.

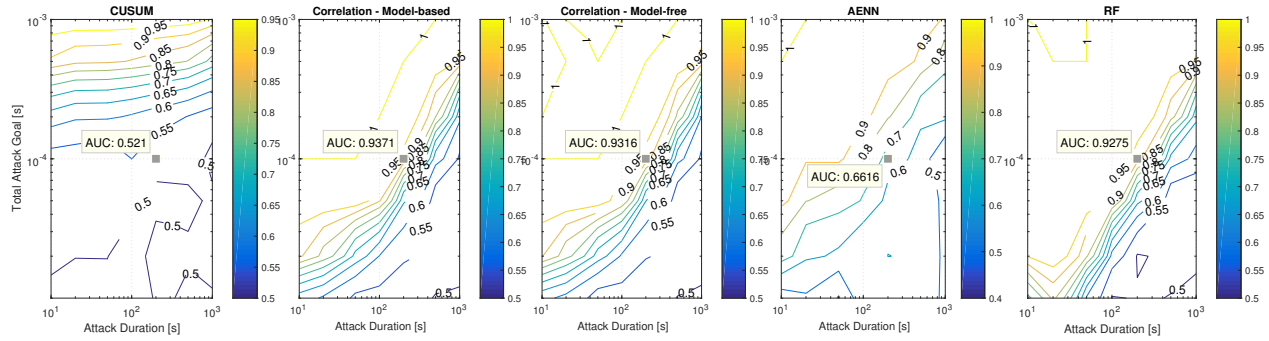
Additionally, the performance of CUSUM is much inferior to all the proposed correlation-inspired detectors for both clocks, especially for clock B, generalizing the observation from Figure 5. On the other hand, the proposed detectors perform generally well even for relatively low-rate attacks (low $\mu_a(t), \Delta\tau^a$, and high Δt^a). This shows the importance of our hypothesis that the relation between A and ΔO_m can play an important role in the detection of TSAs. Besides, the fact that CUSUM totally fails in detecting attacks against clock B could be explained by that clock B is not very stable, which makes it hard for CUSUM to avoid false positives. On the other hand, the proposed detectors do not suffer from this problem as they not only depend on the adjustment sequences, but also on the phasor measurements.

Furthermore, we can observe that among the 4 proposed detectors, the performance of AENN is clearly inferior to other detectors. Remember that AENN was only trained on non-attacked data, and thus only learns a good representation of what it sees, lacking an idea of how an attack might affect the data. On the other hand, the remaining proposed detectors had apriori knowledge about the attack, in the form of training attacked data (RF detector), or in the form of knowledge that an attack would change the correlation (model-based and model-free detectors).

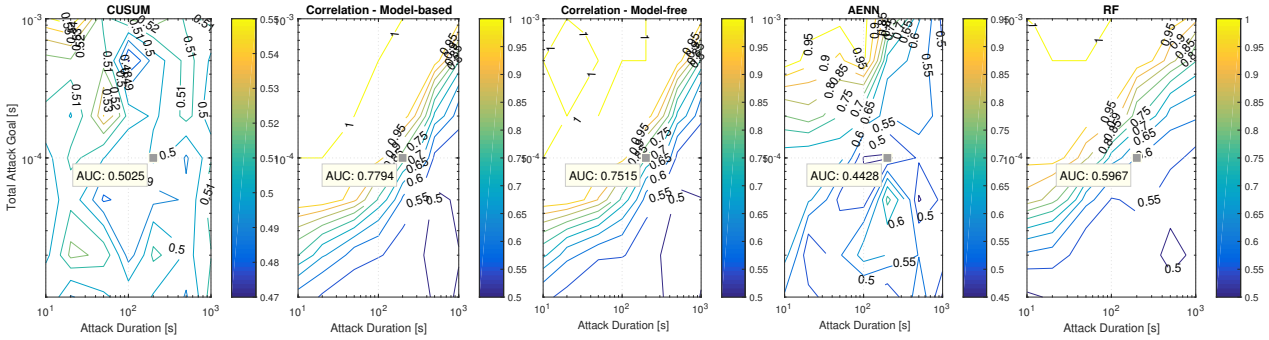
When comparing the model-based and the model-free detectors with the RF detector, we see that their performance is very close. For clock A, the RF detector performs slightly better for most of the attack scenarios, unlike for the scenario considered in Figure 5 (where the model-based and the model-free detectors outperformed the RF detector). However for clock B, the proposed correlation-based detectors (model-based and model-free) perform slightly better, which demonstrates the sensitivity of the correlation w.r.t. attacks. In general, one would expect the RF detector to perform better since it uses much more data (attacked and non-attacked) to learn how to differentiate between attacked and non-attacked behavior. Meanwhile, the two correlation-based detectors depend only on the correlation sequences and on the hypothesis that the correlation value will change due to an attack. Judging from the results, it seems like for some scenarios (mostly for clock B), the information in the correlation values (used in the two correlation-based detectors) is more useful for distinguishing between attacked and non-attacked behaviors than the information in the original raw data (used in the RF detector).

Finally, the plots also show that even though we simulate a PI-Controller, the performance of the model-based detector is comparable to the model-free detector, as long as σ^* and θ could be estimated accurately. In general, we can conclude that the simple correlation-based detectors (model-based and model-free) perform very well, especially for less stable clocks. They are also superior to other detectors regarding their low computational complexity, especially compared to computationally extensive machine learning algorithms.

The effect of the chosen attack strategy on the detection performance is shown in Figure 7. Due to lack of space,



(a) Clock A



(b) Clock B

Fig. 6: AUC isoquants for the different detectors, for (a) clock A and (b) clock B.

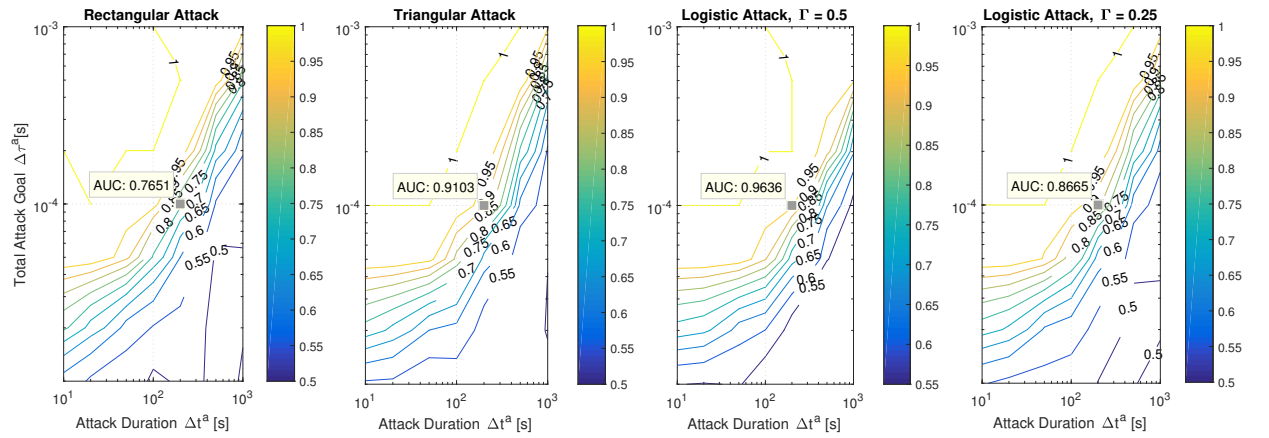


Fig. 7: AUC isoquants for the model-based detector when implementing different attack strategies on clock B.

we show only results for the model-based detector on clock B, but similar behavior can be observed for the other three proposed detectors and for clock A. The results show that the rectangular attack is slightly harder to detect than the other attack strategies. For example, the annotations showing the AUC at $\Delta t^a = 200s, \Delta\tau^a = 100\mu s$ show that the rectangular attack strategy is significantly harder to detect than the other strategies for this dataset, which might seem surprising. One reason for this is that all other strategies start and end the attack with lower values of the attack rate $\mu_a(t)$, which means that μ_a^{max}, and hence the detection probability will be higher. Another reason for this observation is that an attack might only be detected when $\mu_a(t)$ is changing. When $\mu_a(t)$ is constant, an attack cannot be distinguished from a non-malicious PMU that has a different initial frequency deviation $\gamma(0)$. While $\mu_a(t)$ is only changing at t_s^a and t_e^a in the rectangular attack, it is continuously changing in the intervals $t \in [t_s^a, t_1^a[$ and $t \in [t_2^a, t_e^a]$ in the other attack strategies, and thus increasing the detection probability. However, we can conclude in general that there is no significant difference in performance depending on the implemented attack strategy, and therefore our four proposed detectors are robust to a wide variety of attacks.

Finally, the results show that the model-based and the

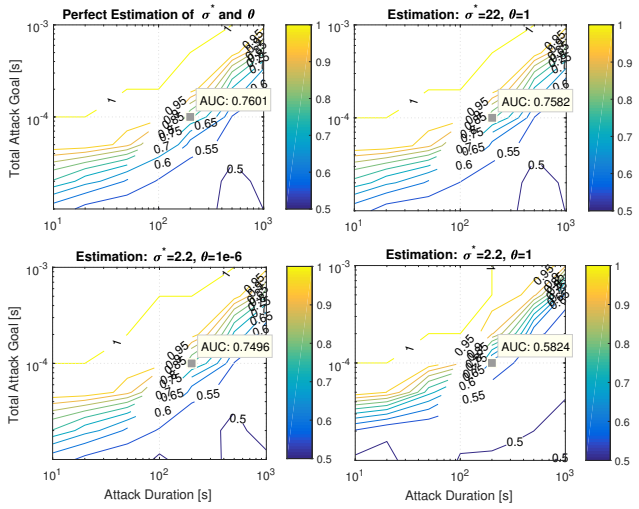


Fig. 8: AUC isoquants for the model-based detector under different mis-estimation levels of σ^* and θ for clock B.

model-free detectors performed equally well. Recall that the model-free detector requires access to historical non-attacked data in order to estimate the expected value of the correlation. Therefore, if access to such data is limited, it would be better to use the model-based detector. However if historical non-attacked data is available, the comparison between the two detectors would depend on the accuracy of the estimation of σ^* and θ for the model-based detector. Therefore, in what follows we evaluate the sensitivity of the model-based detector to a potential mis-estimation of σ^* or θ, and hence $f_\rho(t, \sigma^*, K_p, \theta)$. For this, we considered the rectangular attack strategy on clock B ($\sigma^* = 22, \theta = 10^{-6}$), and simulated different mis-estimation levels: (1) only for σ^* ($\sigma^* = 2.2, \theta = 10^{-6}$), (2) only for θ ($\sigma^* = 22, \theta = 1$), and both ($\sigma^* = 2.2, \theta = 1$). Figure 8 shows the detection performance with and without mis-estimation. From the figure and the annotations at ($\Delta t^a = 200$s and $\Delta\tau^a = 100\mu$s) we can see that the performance degradation is quite insignificant when only one of the two model parameters is mis-estimated. However, a fairly significant degradation can be observed when both parameters are mis-estimated (bottom right plot). Therefore, the model-free detector may provide a better alternative when it is hard to estimate σ^* and θ.

VII. Conclusions

In this paper we proposed a novel approach for detecting time synchronization attacks on PMUs, leveraging the interaction between the time synchronization system and the power system. We provided a model for the measured phase angle by a PMU. We then used the model to obtain a closed-form expression of the correlation between the frequency adjustments implemented by the clock and the resulting change in the measured phase angle. The closed-form expression was shown to be accurate and valid for a wide variety of realistic clocks. Furthermore, we proposed four time synchronization attack detectors; two threshold-based and two machine-learning based detectors, all exploiting the intuition that an attack would change the correlation value. We evaluated the proposed detectors, under different attack strategies and targets and showed that they outperform state-of-the-art clock anomaly detection. Our results also show that low-complexity threshold-based detectors could achieve comparable performance to computationally extensive machine learning-based detectors, even for low-rate TSAs.

Acknowledgement

This work was partly funded by the Swedish Civil Contingencies Agency (MSB) through the CERCES project, by the Swedish Foundation for Strategic Research (SSF) through the CLAS project, and by the European Institute of Innovation and Technology (EIT). This body of the European Union receives support from the European Unions Horizon 2020 research and innovation programme.

References

[1] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2008*, pp. 1–300, July 2008.

[2] X. Jiang, J. Zhang, B. Harding, J. Makela, and A. Dominguez-Garcia, "Spoofing GPS receiver clock offset of phasor measurement units," *IEEE Trans. on Power Systems*, vol. 28, no. 3, pp. 3253–3262, Aug 2013.

[3] E. Itkin and A. Wool, "A security analysis and revised security extension for the precision time protocol," in *Proc. of IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, Sept 2016, pp. 1–6.

[4] S. Barreto, A. Suresh, and J. LeBoudec, "Cyber-attack on Packet-Based time synchronization protocols: the undetectable delay box," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Taipei, Taiwan, May 2016.

[5] E. Shereen, F. Bitard, G. Dán, S. Fries, and T. Sel, "Next steps in security for time synchronization: Experiences from implementing IEEE 1588 v2.1," in *Proc. of IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, Sep. 2019.

[6] M. S. Almas, L. Vanfretti, R. S. Singh, and G. M. Jonsdottir, "Vulnerability of synchrophasor-based wampac applications to time synchronization spoofing," *IEEE Trans. on Smart Grid*, pp. 1–1, 2017.

[7] S. Barreto, M. Pignati, G. Dán, J. L. Boudec, and M. Paolone, "Undetectable timing-attack on linear state-estimation by using rank-1 approximation," *IEEE Trans. on Smart Grid*, vol. 9, no. 4, pp. 3530–3542, July 2018.

[8] L. A. van Mastrigt, A. J. van der Wal, and P. J. Oonincx, "Exploiting the doppler effect in gps to monitor signal integrity and to detect spoofing," in *2015 International Association of Institutes of Navigation World Congress (IAIN)*, Oct 2015, pp. 1–8.

[9] C. H. Kang, S. Y. Kim, and C. G. Park, "Adaptive complex-EKF-based DOA estimation for GPS spoofing detection," *IET Signal Processing*, vol. 12, no. 2, pp. 174–181, 2018.

[10] E. Nunzi, L. Galleani, P. Tavella, and P. Carbone, "Detection of anomalies in the behavior of atomic clocks," *IEEE Trans. on Instrumentation and Measurement*, vol. 56, no. 2, pp. 523–528, Apr 2007.

[11] V. Formichella and P. Tavella, "A recursive clock anomalies detector with double exponential smoothing," in *Proc. of European Frequency and Time Forum (EFTF)*, April 2018, pp. 245–248.

[12] K. Dieter, B. McCamish, E. Cotilla-Sanchez, R. B. Bass, S. Wallace, and X. Zhao, "Power system spoof detection with a hybrid hardware/software benchmarking framework," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, Aug 2018, pp. 1–5.

[13] J. Valenzuela, J. Wang, and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1052–1062, May 2013.

[14] S. Mousavian, J. Valenzuela, and J. Wang, "Real-time data reassurance in electrical power systems based on artificial neural networks," *Electric Power Systems Research*, vol. 96, pp. 285 – 295, 2013.

[15] E. S. PAGE, "Continuous Inspection Schemes," *Biometrika*, vol. 41, no. 1-2, pp. 100–115, Jun 1954. [Online]. Available: https://doi.org/10.1093/biomet/41.1-2.100

[16] S. Bhamidipati, T. Y. Mina, and G. X. Gao, "Gps time authentication against spoofing via a network of receivers for power systems," in *Proc. of IEEE/ION Position, Location and Navigation Symposium (PLANS)*, April 2018, pp. 1485–1491.

[17] E. Lisova, E. Uhlemann, W. Steiner, J. kerberg, and M. Bjrkman, "Game theory applied to secure clock synchronization with ieee 1588," in *Proc. of IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, Sep. 2016.

[18] A. Abur and A. Exposito, *Power system state estimation: theory and implementation*. CRC, 2004, vol. 24.

[19] M. Yasinzadeh and M. Akhbari, "Detection of pmu spoofing in power grid based on phasor measurement analysis," *IET Generation, Transmission Distribution*, vol. 12, no. 9, pp. 1980–1987, 2018.

[20] X. Fan, L. Du, and D. Duan, "Synchrophasor data correction under gps spoofing attack: A state estimation-based approach," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4538–4546, Sep. 2018.

[21] S. B. Andrade, J. L. Boudec, E. Shereen, G. Dán, M. Pignati, and M. Paolone, "A continuum of undetectable timing-attacks on pmu-based linear state-estimation," in *IEEE Intl. Conf. on Smart Grid Communications (SmartGridComm)*, Oct 2017, pp. 473–479.

[22] E. Shereen, M. Delcourt, S. Barreto, G. Dán, J. L. Boudec, and M. Paolone, "Feasibility of time synchronization attacks against pmu-based state-estimation," *IEEE Transactions on Instrumentation and Measurement*, to appear.

[23] E. Shereen and G. Dán, "Correlation-based detection of pmu time synchronization attacks," in *IEEE Intl. Conf. on Smart Grid Communications (SmartGridComm)*, Oct 2018.

[24] R. Zhang, F. Hflinger, and L. Reindl, "Tdoa-based localization using interacting multiple model estimator and ultrasonic transmitter/receiver," *IEEE Trans. on Instrumentation and Measurement*, 2013.

[25] R. G. Lins, S. N. Givigi, and P. R. G. Kurka, "Vision-based measurement for localization of objects in 3-d for robotic applications," *IEEE Trans. on Instrumentation and Measurement*, 2015.

[26] L. Zanni, S. Sarri, M. Pignati, R. Cherkaoui, and M. Paolone, "Probabilistic assessment of the process-noise covariance matrix of discrete Kalman filter state estimation of active distribution networks," in *2014 Intl. Conf. on Probabilistic Methods Applied to Power Systems (PMAPS)*, Jul 2014, pp. 1–6.

[27] E. Bibbona, G. Panfilo, and P. Tavella, "The Ornstein-Uhlenbeck process as a model of a low pass filtered white noise," *Metrologia*, vol. 45, p. S117, Dec 2008.

[28] G. Maruyama, "Continuous markov processes and stochastic equations," *Rend Circ Math Palermo*, 1955.

[29] K. Correll, N. Barendt, and M. Branicky, "Design considerations for software only implementations of the IEEE 1588 precision time protocol," in *Conf. on IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, 2006.

[30] M. Markou and S. Singh, "Novelty detection: a review - part 2:: neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499 – 2521, 2003.

[31] T. K. Ho, "Random decision forests," in *Proceedings of 3rd Intl. Conf. on Document Analysis and Recognition*, vol. 1, Aug 1995, pp. 278–282.

[32] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[33] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006.